**Computer Vision Project Phase 3 – Group 7**

**Industry Partner – AVO Insights**

**Plane Detection**

The zipped file consists of 5 python files that computes and calculates values for plane tracking and detection.

*A short summary of the 5 files are as follows:*

1. Common.py – this file contains some of the common methods used by several classes to work as a whole. Some common functionalities like drawing a rectangle for bounding box, draw keypoints for features, normalization, etc.
2. Tst\_scene\_render.py – this file contains the class for rendering frames onto the screen. A main method renders the output on to the screen frame by frame.
3. Video.py – this file contains classes for object rendering passed through dictionary and defines presets for the same.
4. Plane\_tracking.py – this file extracts features from frames using the ORB algorithm and FlannBasedknnMatching, and RANSAC to pass the nearest neighbors of extracted features to a dictionary.
5. Plane\_ar.py – this file draws utilizes the features extracted from plane\_tracker.py and draws a basic ‘house’ (Simple model of a house - cube with a triangular prism "roof") hardcoded from arrays representing x-y coordinates.

**Instructions on running the code:**

Windows:

* Open GitBash
* Change directory to the folder containing all the files
* Type “python plane\_ar.py <video file>”
* An output window pops up.
* Use the mouse to draw a bounding box around a plane of your choice
* Press spacebar to pause, and ‘c’ to clear anchor points and draw a new bounding box.
* Use 'focal' slider to adjust to camera focal length for proper video augmentation.

Linux:

* Open the terminal
* Change directory to the folder containing all the files
* Type “python plane\_ar.py <video file>”
* An output window pops up.
* Use the mouse to draw a bounding box around a plane of your choice
* Press spacebar to pause, and ‘c’ to clear anchor points and draw a new bounding box.
* Use 'focal' slider to adjust to camera focal length for proper video augmentation.

MacOS:

* Open the terminal
* Change directory to the folder containing all the files
* Type “python plane\_ar.py <video file>”
* An output window pops up.
* Use the mouse to draw a bounding box around a plane of your choice
* Press spacebar to pause, and ‘c’ to clear anchor points and draw a new bounding box.
* Use 'focal' slider to adjust to camera focal length for proper video augmentation.

***Results obtained:***

A video file named “DemoScreenRecord” demonstrates plane tracking and pinning an AR Based Object in a video, given a user input within a bounding box. The code developed performs well in tracking the plane within a bounding box by extracting the features and dropping anchor points to track these features within each frame of the entire video.

In the second half of the video, we also demonstrate that the model house is stable as long as the features extracted are intact. (As seen in the video, once a slice of the bread is cut, the algorithm loses a few extracted features and anchor points, hence making the model house unstable).

***Limitations Encountered:***

The video file named “LimitationsScreenRecord” demonstrates the limitation encountered – glare within a video. Glaring videos mean that the nearest neighbors are incorrectly computed making it hard for the algorithm to drop anchor points to track.

As seen in the video, the top plane of the google mini box is constantly fidgety and struggles to drop anchor points that are stable, which it can use to track. Hence, the simple model of the house is unstable.

However, a bounding box drawn on the side of the google mini box is able to build is a stable model of the house since it is not directly under light and does not experience glare.